

What Is Claimed Is:

1 1. A method for dynamically configuring selected methods for
2 instrument-based profiling at run-time, comprising:
3 identifying a root method in a target application, wherein only methods
4 that are reachable from the root method during execution of the target application
5 are to be instrumented;
6 instrumenting the root method after it is identified; and
7 upon executing a given instrumented method, determining if the given
8 instrumented method is about to be executed for the first time, and if so,
9 instrumenting methods that,
10 are called by the given instrumented method,
11 are loaded, and
12 have not been instrumented before.

1 2. The method of claim 1, wherein identifying the root method
2 involves allowing a user to specify the root method.

1 3. The method of claim 1, wherein determining if the given
2 instrumented method is about to be executed for the first time involves executing
3 instrumentation code within the given instrumented method, wherein the
4 instrumentation code is executed whenever the given instrumented method is
5 called, and is executed before any other instructions of the given instrumented
6 method are executed.

1 4. The method of claim 3,

2 wherein the instrumentation code checks a global executed-once-or-more
3 flag associated with the given instrumented method, which is initially set to false;
4 wherein if the executed-once-or-more flag is false, the instrumentation
5 code knows that the given instrumented method has not been called before, so it
6 performs instrumentation operations as necessary and sets the instrumented once
7 or more flag to true; and
8 wherein if the executed-once-or-more flag is true is true, the
9 instrumentation code does not perform instrumentation operations.

1 5. The method of claim 1, wherein if a call to a virtual method is
2 encountered in a given instrumented method that is about to be executed for the
3 first time, the method further comprises:
4 identifying a class for the virtual method based upon an object type
5 associated with the call site; and
6 instrumenting methods corresponding to the virtual method in the
7 identified class and in associated subclasses.

1 6. The method of claim 5, wherein if the identified class does not
2 have an implementation of the virtual method, the method additionally involves
3 identifying a nearest superclass of the identified class that has an implementation
4 of the virtual method.

1 7. The method of claim 1, wherein instrumenting a method involves
2 dynamically patching the method while the target application is executing.

1 8. The method of claim 1, wherein instrumenting a method involves
2 inserting profiling instrumentation code into the method, wherein the profiling
3 instrumentation code includes:
4 method entry code that takes a first time measurement at the beginning of
5 a method;
6 method exit code that takes a second time measurement at the end of the
7 method; and
8 wherein the first and second time measurements are used to calculate an
9 execution time for the method.

1 9. The method of claim 8, wherein the method entry code determines
2 if the given instrumented method is about to be executed for the first time.

1 10. The method of claim 1, wherein the tasks of identifying methods
2 and instrumenting methods are performed by a remote profiler client that
3 communicates with a virtual machine executing the target application.

1 11. The method of claim 1, wherein code that makes up the target
2 application includes platform-independent Java bytecodes.

1 12. A computer-readable storage medium storing instructions that
2 when executed by a computer cause the computer to perform a method for
3 dynamically configuring selected methods for instrument-based profiling at run-
4 time, the method comprising:
5 identifying a root method in a target application, wherein only methods
6 that are reachable from the root method during execution of the target application
7 are to be instrumented;

8 instrumenting the root method after it is identified; and
9 upon executing a given instrumented method, determining if the given
10 instrumented method is about to be executed for the first time, and if so,
11 instrumenting methods that,
12 are called by the given instrumented method,
13 are loaded, and
14 have not been instrumented before.

1 13. The computer-readable storage medium of claim 12, wherein
2 identifying the root method involves allowing a user to specify the root method.

1 14. The computer-readable storage medium of claim 12, wherein
2 determining if the given instrumented method is about to be executed for the first
3 time involves executing instrumentation code within the given instrumented
4 method, wherein the instrumentation code is executed whenever the given
5 instrumented method is called, and is executed before any other instructions of the
6 given instrumented method are executed.

1 15. The computer-readable storage medium of claim 14,
2 wherein the instrumentation code checks a global executed-once-or-more
3 flag associated with the given instrumented method, which is initially set to false;
4 wherein if the executed-once-or-more flag is false, the instrumentation
5 code knows that the given instrumented method has not been called before, so it
6 performs instrumentation operations as necessary and sets the instrumented once
7 or more flag to true; and
8 wherein if the executed-once-or-more flag is true is true, the
9 instrumentation code does not perform instrumentation operations.

1 16. The computer-readable storage medium of claim 12, wherein if a
2 call to a virtual method is encountered in a given instrumented method that is
3 about to be executed for the first time, the method further comprises:
4 identifying a class for the virtual method based upon an object type
5 associated with the call site; and
6 instrumenting methods corresponding to the virtual method in the
7 identified class and in associated subclasses.

1 17. The computer-readable storage medium of claim 16, wherein if the
2 identified class does not have an implementation of the virtual method, the
3 method additionally involves identifying a nearest superclass of the identified
4 class that has an implementation of the virtual method.

1 18. The computer-readable storage medium of claim 12, wherein
2 instrumenting a method involves dynamically patching the method while the
3 target application is executing.

1 19. The computer-readable storage medium of claim 12, wherein
2 instrumenting a method involves inserting profiling instrumentation code into the
3 method, wherein the profiling instrumentation code includes:
4 method entry code that takes a first time measurement at the beginning of
5 a method;
6 method exit code that takes a second time measurement at the end of the
7 method; and
8 wherein the first and second time measurements are used to calculate an
9 execution time for the method.

1 20. The computer-readable storage medium of claim 19, wherein the
2 method entry code determines if the given instrumented method is about to be
3 executed for the first time.

1 21. The computer-readable storage medium of claim 12, wherein the
2 tasks of identifying methods and instrumenting methods are performed by a
3 remote profiler client that communicates with a virtual machine executing the
4 target application.

1 22. The computer-readable storage medium of claim 12, wherein code
2 that makes up the target application includes platform-independent Java
3 bytecodes.

1 23. An apparatus for dynamically configuring selected methods for
2 instrument-based profiling at run-time, comprising:
3 an identification mechanism configured to identify a root method in a
4 target application, wherein only methods that are reachable from the root method
5 during execution of the target application are to be instrumented; and
6 an instrumentation mechanism configured to instrument the root method
7 after it is identified;
8 wherein when a given instrumented method is executed, the
9 instrumentation mechanism is configured to determine if the given instrumented
10 method is about to be executed for the first time, and if so, to instrument methods
11 that,
12 are called by the given instrumented method,
13 are loaded, and

14 have not been instrumented before.

1 24. The apparatus of claim 23, wherein the identification mechanism is
2 configured to identify the root method by allowing a user to specify the root
3 method.

1 25. The apparatus of claim 23, wherein while determining if the given
2 instrumented method is about to be executed for the first time, the instrumentation
3 mechanism is configured to execute instrumentation code within the given
4 instrumented method, wherein the instrumentation code is executed whenever the
5 given instrumented method is called, and is executed before any other instructions
6 of the given instrumented method are executed.

1 26. The apparatus of claim 25,
2 wherein the instrumentation code checks a global executed-once-or-more
3 flag associated with the given instrumented method, which is initially set to false;
4 wherein if the executed-once-or-more flag is false, the instrumentation
5 code knows that the given instrumented method has not been called before, so it
6 performs instrumentation operations as necessary and sets the instrumented once
7 or more flag to true; and
8 wherein if the executed-once-or-more flag is true is true, the
9 instrumentation code does not perform instrumentation operations.

1 27. The apparatus of claim 23, wherein if a call to a virtual method is
2 encountered in a given instrumented method that is about to be executed for the
3 first time, the instrumentation mechanism is configured to:

4 identify a class for the virtual method based upon an object type associated
5 with the call site; and to
6 instrument methods corresponding to the virtual method in the identified
7 class and in associated subclasses.

1 28. The apparatus of claim 27, wherein if the identified class does not
2 have an implementation of the virtual method, the instrumentation mechanism is
3 configured to identify a nearest superclass of the identified class that has an
4 implementation of the virtual method.

1 29. The apparatus of claim 23, wherein the instrumentation mechanism
2 is configured to instrument a method by dynamically patching the method while
3 the target application is executing.

1 30. The apparatus of claim 23, wherein the instrumentation mechanism
2 is configured to instrument a method by inserting profiling instrumentation code
3 into the method, wherein the profiling instrumentation code includes:
4 method entry code that takes a first time measurement at the beginning of
5 a method;
6 method exit code that takes a second time measurement at the end of the
7 method; and
8 wherein the first and second time measurements are used to calculate an
9 execution time for the method.

1 31. The apparatus of claim 30, wherein the method entry code
2 determines if the given instrumented method is about to be executed for the first
3 time.

1 32. The apparatus of claim 23, wherein the identification mechanism
2 and the instrumentation mechanism are located within a remote profiler client that
3 communicates with a virtual machine executing the target application.

1 33. The apparatus of claim 23, wherein code that makes up the target
2 application includes platform-independent Java bytecodes.